1. Why are functions advantageous to have in your programs?

Ans: 1) Functions help in re-usability of code and hence restricts duplication of code.

2) Functions helps in breaking a large functionality into pieces.

3) They help to better understand the code by reducing the complexity of the program.

4) Their implementation reduces debugging time as they enhance the clarity of the program.

2. When does the code in a function run: when it's specified or when it's called?

Ans. The code in the function is executed or run when the function is called in the program.

3. What statement creates a function?

Ans. 'def' keyword is used to create or define a function.

For ex → def test():

```
            print(1+2)
```

4. What is the difference between a function and a function call?

Ans. A function contains the complete definition of a function. It is defined by the def keyword. Function means the whole internal working code of the function.

For ex: def test(x):

```
            print(x+2)
```

The above code shows the internal code of a function.

But a function call statement calls the function for execution. Function call statement means to refer the function that is written in some other location.

For ex: test(3)

5. How many global scopes are there in a Python program? How many local scopes?

Ans. There 4 scopes in Python:

1) Built -in Scope: It is the widest scope in Python. All reserved keywords fall under this scope. We can call the keywords any where in the program without defining them before. Ex. False, True, for, finally, is, lambda etc.
2) Global Scope: The variables defined with this scope is available all through the code. These are defined in the main body of the program.
3) Enclosing Scope: This scope exists for the nested function. Variables defined with this scope are available to the enclosing function and the inner functions.
4) Local Scope: Variable with this scope are available only to the inner function. These variables are not recognized outside the function they are defined in.

```
Example: a=0                        #Global Scope
         def test1():
             sum=0                  #Enclosing Scope
             def hello():
                     add=12         #Local Scope
```

6. What happens to variables in a local scope when the function call returns?

Ans. The memory occupied by Local variable is freed up and made available for other variables or we can say that local variables are destroyed once the control flow jumps to the other part of the program after executing return statement in a function.

7. **What is the concept of a return value? Is it possible to have a return value in an expression?**

Ans. Return is used to go back to the function calling statement after completing the task or whenever it appears in the code and continue the normal flow of the program. Yes, it is possible to have return value in expression.

Ex.
```
def test():
        return 123

a= test()
print(a)
output: 123
```

If the function does not return a value then a Nonetype value None is assigned to the variable.

8. **If a function does not have a return statement, what is the return value of a call to that function?**

Ans. Return Value of a call to a function which has no return statement: **None**

9. **How do you make a function variable refer to the global variable?**

Ans. By using 'global' keyword

10. **What is the data type of None?**

Ans. NoneType

11. **What does the sentence import areallyourpetsnamederic do?**

Ans. Import function searches for the areallyourpetsnamederic.py module and then binds the results of that search to a name in the local scope. The value returned by the function is then reflected in the output of the initial code.

12. **If you had a bacon() feature in a spam module, what would you call it after importing spam?**

Ans. spam.bacon()

13. **What can you do to save a programme from crashing if it encounters an error?**

Ans. We can perform Exception Handling to prevent our program from crashing.

14. **What is the purpose of the try clause? What is the purpose of the except clause?**

Ans. Try clause encloses the code which is suspicious of errors. Except clause is used to handle the error. When ever error occur in the try block then our program control jumps to the except block and handles the error.